

Cloudbased Production Optimization – Potential and Limits Today

Roger Feist^{1,a*}

¹Siegener Straße 152, 57223 Kreuztal, Germany

^ar.feist@achenbach.de

Keywords: Big Data, Analytics, Production, Cloud, Google, Big Query, NoSQL, Optimization

Abstract. This paper shows how a databased approach towards production optimization was realized with the help of cloud-technologies. Several uncertainties, either in the manufacturing of the producing machines or in the production on these machines can be systematically reduced. In this way a significant improvement in production amount, but also in produced quality can be reached.

Introduction

In spite of considerable experience and knowledge and in spite of numerous physical models a significant amount of uncertainty related to the result of the production process is still inherent in the field of plastic metal working. This applies to both, the design of the machine executing the deformation-process and the resulting product itself. The reasons for these uncertainties are numerous. In many cases however they result from insufficient knowledge of the real boundary-conditions. The producer of machinery usually has only partly knowledge of the range of products, which are rolled on the machine during daily production.

When looking on plastic deformation processes, especially in foil rolling mills, tribological and thermal effects have a high influence [1][2]. In these cases the relevant physical parameters like oil-viscosity are not possible to measure ‘inline’ with reasonable efforts and they are continuously changing. This means that ‘offline’-analysis in a laboratory can help to understand problems but cannot help solving it, because of the delays in the lab the result of the analysis will available too late to change the process.

The continuous usage of actual, measurable process-values containing traces of the relevant parameters can serve as a loophole from this trap. If these values are recorded and evaluated over a sufficiently long timespan, either a prediction of the production result can be directly made or the unknown parameters in an existing physical model can be identified making use of a parallel running optimizing-algorithm.

A critical factor for the identification strategy described above is the availability of the computational result within a mission-critical timeframe. The timeframe depends on the respective task and can be up to some minutes, but data acquisition and processing must be included.

At the background of the huge number of relevant values and the high requirements on computational power the usage of classic ‘on premise’ architectures is strongly limited. This limitation can be overcome by using cloud based services, because the big amounts of computational power and memory necessary during the learning-process can be provided in an economical way by means of the almost infinite scalability.

For the validation of decisions made in the design process of the producing machine the production data have similar valuable content. They can be used to verify the design-models or to adapt when necessary. If newly built machines vary in their specification from the ones the data are from, it is essential to have a broad database to cover variation in the machines and tasks.

Requirements on the Data Acquisition

Even if the focus is only on data belonging to a single production step it is notable, that the relevant data are generated in different, strongly heterogeneous systems. At a single machine normally a number of control systems from different suppliers are installed, where each single system can be implemented by different suppliers. For example, relevant data for a CNC-drilling machine data can also come from a robot feeding the machine. For the recording and analysis of a production step it is essential to have an acquisition-system with a good connectivity to make the related data accessible, independent from the control platform they are generated on.

In contrast to many other branches in the metals-industry recording systems, capable to register a high number of process values with a resolution of less than a millisecond are already widely used. Therefore the ‘connectivity-task’ is solved in principal. The data collected by these systems can be used for quality reports or for finding root-causes of an incident as long as the focus is only on one machine. When the focus is on the comparison or correlation between results of different production steps, these systems are tightly limited because of the network structure.

When looking at a classic maintenance scenario which is to find the root cause of an incident in the machine it is evident, that this task can only be solved, if the relevant data is available. Otherwise every result of analysis remains only an assumption. As an incident is always an unexpected event that was not foreseen, it is clear, that also the choice of relevant data necessary to analyze the event, cannot be foreseen either. If the choice of recorded data is restricted too much, a situation in which relevant, for the further analysis necessary data is missing will occur frequently. An important requirement on the data acquisition system is therefore the ability to collect a large selection of attributes.

In the transition from “reactive maintenance” to predictive algorithms the incident itself becomes predictable, but under similar preconditions. If a supervised approach [3] is used it is mandatory to have recorded incidents together with the relevant data. Initially the situation regarding the choice of data is the same as in the reactive case: Neither the incident to be predicted nor the relevant attributes are known. Thus also here a broad selection of recorded attributes is necessary.

The intended use of an unsupervised approach creates an additional requirement on the system: Next to the data relevant for a given incident these algorithms also need data to identify the normal case. From changes against the statistically identified ‘ok’-state the derivation of the ‘not ok’-state will take place. Therefore it is important to collect, next to a wide choice of attributes – the ‘width of the recorded table’ – over a sufficiently long time ‘depth of the recorded table’. These two last requirements applied to industrial cases lead to the requirement to apply big-data technologies, as the resulting tables are hardly computable with conventional methods.

In spite of the importance to use historic data for gathering results, a huge historic database is not sufficient to solve most practical tasks. By making use of the historic data the used approaches can be tested and verified, but the application of the findings has always a temporal dimension: A prediction of an event or incident is just helpful in the case, when the prediction occurs sufficiently before the event. Thus data must be analyzed close to their creation.

Another important requirement on the optimization-system comes from the observation of the industrial production process. Products and their failures arise from a sequence of production steps, which may be carried out on different machines and at different locations. To build a comprehensive system for product-optimization it is not enough to analyze only one step of production. The root-cause for a failure in a subsequent production step could be obvious in a set of data created in a preceding step. In these cases an early recognition and avoidance of the failure requires the combination of data of different subsequent production steps inside the analytic system. This requirement involves that the data must be stored in a place outside the single machinery network in an area, which is writable from all related machines in all locations.

Properties of Cloud Based Architectures

Cloud based services provide a number of valuable possibilities for the developer to comply with the requirements in the preceding section. Especially in comparison with an approach that handles the data in a local infrastructure there are many advantages of cloud infrastructure:

Most of the commercial cloud platforms offer largely managed services reducing the administrative efforts necessary to maintain the system to a great extent. For the implementation of the solution presented here the data warehouse 'BigQuery' from Google was chosen, which is a completely managed, replicating NoSQL-Database [4]. The registered user is able to work over a browser-based interface on the data or can upload new data without any installation efforts. BigQuery offers a powerful REST-API allowing the streaming of data into a table or loading it via batch-jobs.

Because of the fact, that the data is replicated internally and distributed on several datacenter, a high degree of availability of the data is guaranteed. The replication saves also computational time needed for complex queries, as the tasks are splitted and run in parallel on different machines. The short response times paired with the ability to store any amount of data, without changes in disk-stacks or hardware, are very helpful for the implementation of the optimization system.

For the application itself the use of 'platform as a service'-services (P.a.a.S.) was preferred, as it enables the system to scale stepwise with the amount of connected machines and users [5]. Services of this kind are allocating the needed CPU-resources for the data handling and to respond to user-requests dynamically. This happens without intervention of the platform-user based on the supervision of latencies in the application. Dependent on the number of connected machines and the number of active users the cloud-based system can scale itself to the right size. Lack of availability or additional works like upgrading computer-systems as they are common in locally hosted systems are avoided on a large scale.

In cases, where it is necessary to run dedicated software on a single server, infrastructure services are used. For these services the user must manage the required resources, but - because of comfortable backup-mechanisms and options to change the used hardware easily by browser-configuration - also here a certain gain in efficiency can be reached compared with local infrastructure [6].

Finally, the users of a cloud-based system gain the worldwide access to the content. After authentication they can use the services usually with the browser interface. If data is hosted in local networks, accessing it usually means the need to implement VPN-connections to other sites of the company or to single user-laptops outside the company-network. If the single user-laptop changes from work inside the company domain and outside, critical safety issues can arise.

In the case of cloud-based data storage even different, independent companies, for example client and supplier can share data without sharing their networks. By separating the credentials for data-access from the credentials for accessing the company-network this necessary scenario for optimization over the vertical chain can be solved.

A critical factor for the realization of a comfortable and at the same time safe system for production optimization is the possibility to authenticate a user or a machine only once inside the system in order to determine his identity. After successful authentication there should be an instance that confirms the identity of the user automatically to all devices. These services, based on the SAML mechanism are available [7], including user and rights-management by different identity providers in the web.

Implementation of the Solution

Based on the described arguments so far the implementation of the data-acquisition and optimization system is consequently using cloud services. The goal was to create a solution capable to collect data in large amounts and make them available near to real-time inside a cloud environment. As the component for the acquisition of data inside the machinery network a

configurable client was implemented on a single board computer. This client can access the data in different PLC-environments. It packs the received data and streams them to the cloud endpoint. Therefore on the machinery network only those memory-resources are needed that enable buffering of the data during temporal unavailability of the internet connection.

The data arriving at the cloud endpoint are checked in terms of integrity, against a number of configurable trigger conditions and then streamed into the data-warehouse. Finally a number of toolkits were integrated in a web-application to ease the work with the data, to generate information and to access the information in a comfortable way. The layout of the dataflow is shown in figure 1.

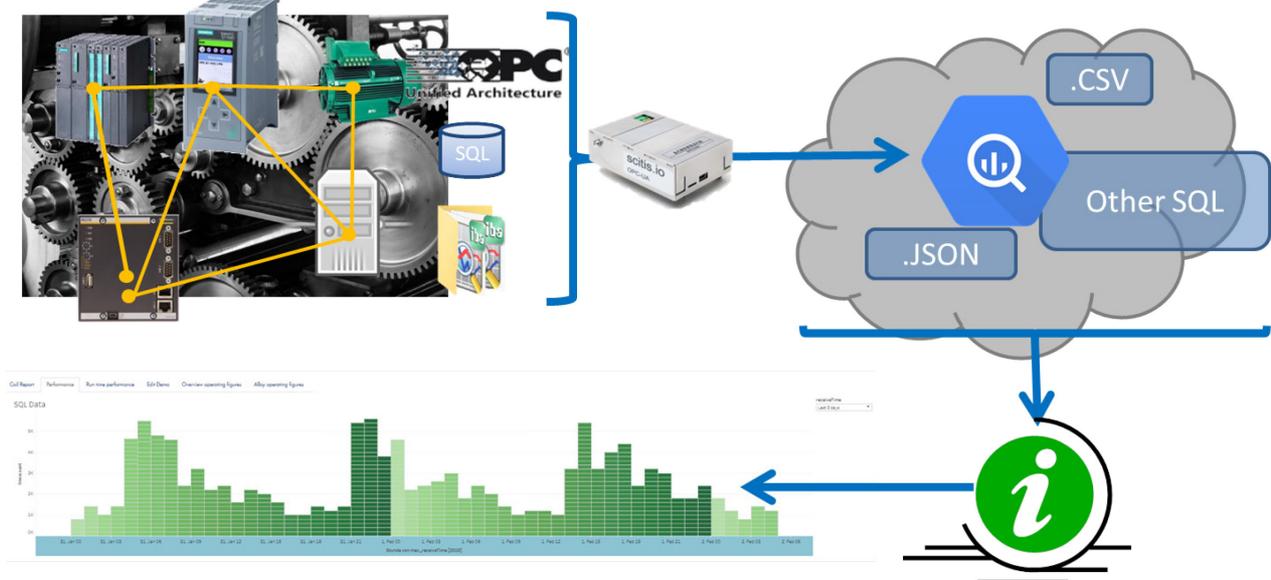


Fig. 1. System Overview

Integration into the machinery network. In order to reach an optimum connectivity into the machinery network the development was concentrated on standard interfaces having the potential to apply in many cases. Here the standard ‘OPC UA’ is the first choice [8]. Almost all modern PLC-systems offer a server-interface according to this standard. Via OPC UA mechanisms the client, in our case the data-acquisition system, can register for receiving arbitrary values from the PLC. With the maximum resolution of 50 milliseconds the client receives complete information about the changes of the registered variables. These time-series of PLC-values are transmitted via HTTPS-telegrams to the cloud-endpoint of the system.

The second implemented standard interface of the acquisition-component is the access to SQL-databases. Many visualization-systems, but also many PLC-systems use methods to write results, e. g. pre-aggregated values, into a SQL-database [9]. One of the most important use-cases of this interface in the metals-area is the access to data generated by the ‘IBA-system’. These systems are commonly used to record data in rolling mills and attached machines to record data with high resolution below one millisecond. In the context of our optimization system, these IBA-systems can be used to pre-aggregate data or to compute the harmonics of a vibration-signal before it is transmitted to the cloud. For example, the spectrum of a vibration sensor placed on a gearbox can be used in combination with the rolling load, torque and rolling speed to assess the criticality of the measured vibration.

The third implemented standard-interface is the possibility to access the file-system inside the machinery network via SMB (Server Message Block Protocol) [10]. If measured data is stored regularly in a standardized structure as files, e. g. in .CSV-files, the data can be read and made available in the cloud.

Data handling in the backend. The cloud-endpoint in the first instance serves as a communication-partner to the devices inside the machinery-network, the ‘Cloudplugs’. It generates request for the re-transmission of faulty data and supervises the quality of the connection. After error-free receipt the data is streamed as a time-series into BigQuery. In parallel to the storage-mechanism the incoming data is sorted and checked for the presence of ‘trigger-events’ as shown in figure 2. Trigger events are configured by the user and serve as conditions to react automatically on specific changes in the recorded values. A simple example for a trigger event is, if a temperature exceeds a given limit, the recognition of an abnormal situation by an AI-function or the change of a production lot.

As reaction on the ignition of a trigger the backend executes a specified action. This action can range from simply sending an e-mail to the person interested or the start of another calculation. One problem with implementing trigger-logic in the system was that the IP-protocols do not guarantee the order of the received telegrams. For evaluating changes of values over time as triggers do it is necessary to have the data complete and ordered by time. To reach this a ‘windowing-algorithm’ was created that delays trigger-evaluations until a certain window of time is completely filled with data.

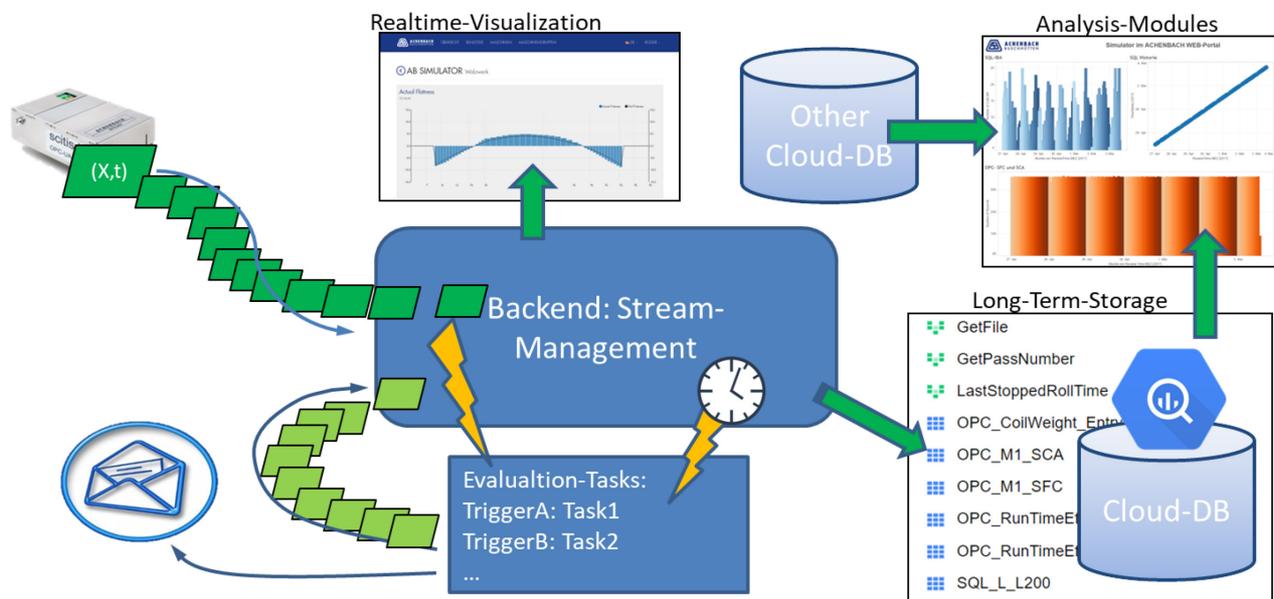


Fig. 2. Backend concept

Frontend. The frontend is designed to offer a comfortable, configurable access to the system’s functions to the user. By using the framework ‘angular.js’ a menu structure was built, that enables on the one hand the execution of supervision- and analytic- tasks and on the other hand the execution of administrative tasks [11]. Part of the administrative tasks is the definition of new connections to machines, the definition of groups of machines and the assignment of user-credentials. The system-admin has support to find errors and can control the remote-distribution of updates to the Cloudplugs. The frontend is implemented using Google AppEngine which makes it highly scalable. An arbitrary number of machines can be connected and an arbitrary number of users can benefit from the created information at the same time.

Expandability and open access structure via REST. An important property of cloud based systems is the expandability. Open WEB-standards ease the integration of third-party software that was used to extend the possibilities of the optimization-tool with relatively low efforts. The approach presented here made use of standard-interfaces to integrate some powerful tools and apply their strengths to the machinery data [12]. In the field of ‘visual data analysis’ the toolbox ‘Tableau’ was chosen as it offers a web-based editor for dashboards and APIs helping to integrate dashboards from the server-package seamlessly into an own website.

In the field of machine-learning (ML) the functions of ‘Rapidminer Server’ were integrated. With this tool it’s possible to create solutions by applying a big choice of learning-algorithms inside a flexible desktop-tool. The completed solution can be published as a server-application. Rapidminer eases the creation of solutions because it offers all algorithms with standardized interfaces and connection points [13]. As an open-source-tool Rapidminer also has a huge worldwide community and it is relatively easy for a specific use case to find a development-partner which is experienced with the software.

As a third option Matlab-functionalities were integrated inside the optimization-portal. Physical models created in Simulink can be called or parametrized from the backend. Matlab offers strong statistics- and ML-toolboxes thus it is an alternative to the use of Rapidminer (figure 3). Critical factor for all integrations was the implementation of efficient interfaces to the production data, which was done either by ODBC-drivers or by BigQueries REST-API.

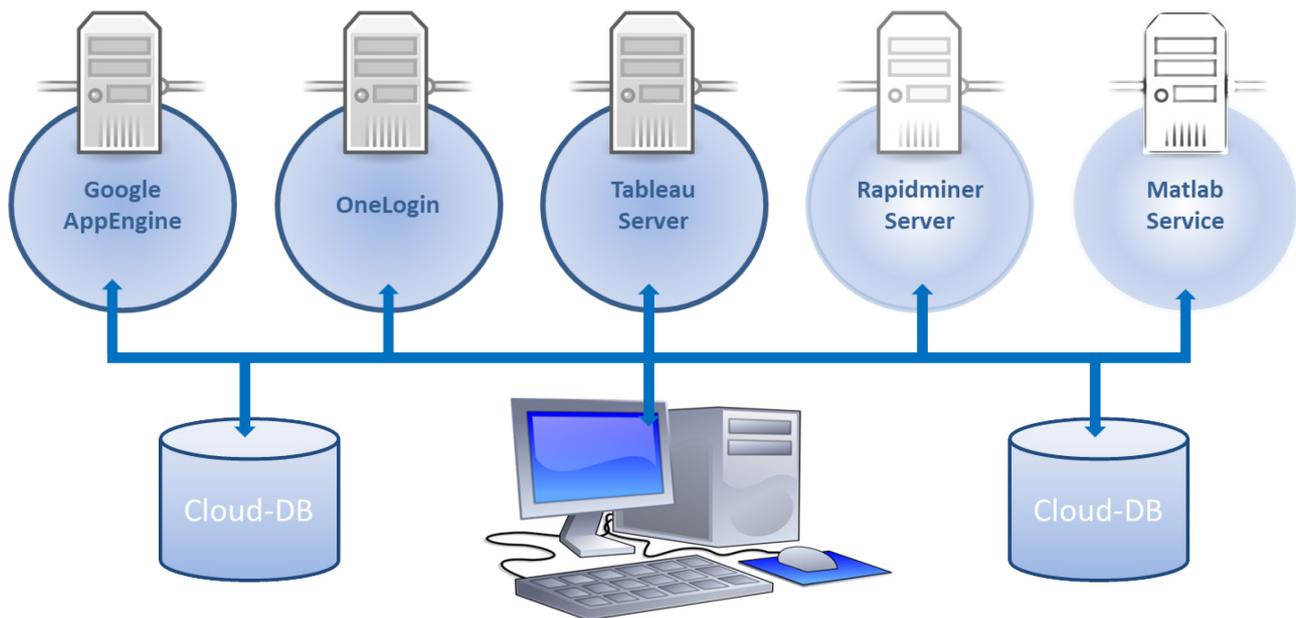


Fig. 3. Toolkit Integration

Results

The resulting system offers a great variety of possibilities. On the machine-side the data of almost all control systems used in the metals-sector can be recorded. This is a result from the high availability of OPC UA-server software in commercial PLC systems. In test-cases PLCs from Siemens, Bachmann and Beckhoff were connected.

It was possible to show, that with today’s common DSL- or glass fiber-connections no noteworthy load on the connection is generated. Because of the chosen packing of the data more than 500 attributes with a resolution of 100 milliseconds can be acquired and transferred into cloud storage. In a test-case it was also shown, that time-series with a resolution down to 1 Millisecond can also be acquired if the application needs such a high resolution. However in most cases where a high sampling of the raw sensor signal was needed, the user was more interested in the frequency-domain than in the time-series. In this case the Fourier-analysis takes place in the machinery network and only a vector containing the spectrum is transmitted to the cloud. This leads to a dataset reminding of a waterfall-diagram which can be analyzed with the techniques of condition-monitoring.

With the trigger-functions described above the supervision of unexpected machine-stops was automated. The responsible persons are informed by email in case of several, configurable events. Because of the web-interface they are always able to receive more information about the background of the downtime.

On Basis of the recorded data a web-based reporting was created (figure 4) and final customers can receive immediately at production time – before the real delivery – the relevant data of the product. This enhances the automated planning of subsequent production steps, because quality and timing are transparent for all participants on the production process.

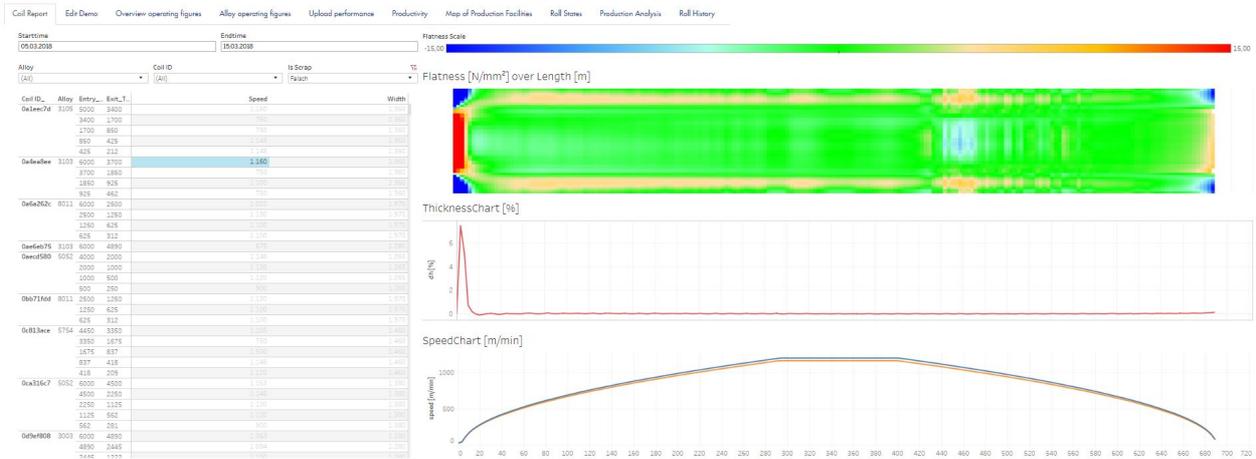


Fig. 4. Coil Report Example

For the cloud-based solution the access to databases of different machines in different locations is no problem, therefore it is possible to monitor and compare critical parameters across locations. In the example (Fig. 5) a world-map with a number of production sites is shown. For each of the sites characteristic values of the productivity can be displayed.

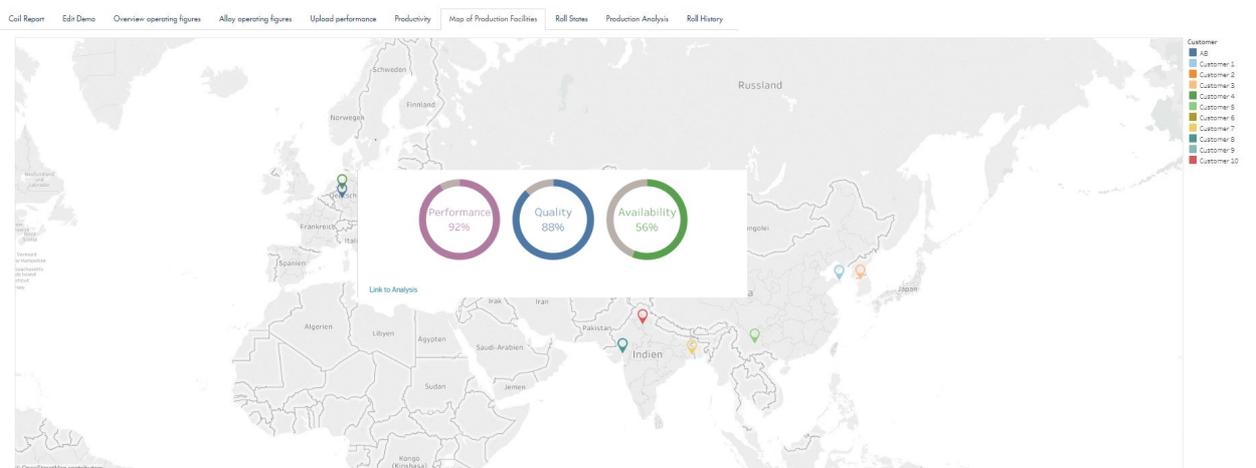


Fig. 5. KPI Supervision

The production of the rolling mills analyzed was always very heterogeneous so far. From the same upstream product different final products were built and the number of different alloys used was high. Therefore it was necessary to sort the different paths through the production-steps by materials, before abnormalities like too long production times, were possible to be detected.

Summary

The basic task to provide access to production data of different machines participating in the production process of rolled metal at different locations was solved by implementing a web-based production portal.

In several use-cases it was shown, that the worldwide internet-infrastructure is appropriate for realizing the underlying concept of a web-based production optimization in industrial scales. By the use of powerful analytic-tools together with high-resolution data new fields of applications can be found. In order to identify normal- and abnormal situations in the production sorting algorithms identifying the production-path were implemented.

The automated recognition of the normal production-path and the ability to inform the producer about abnormal situations is an important target for the further development. By solving this task the machine can directly interfere with the operator, inform him about the abnormalities, and, in simple cases like wrong product-IDs immediately solve the case.

For high frequency data, especially in the case of vibration based condition monitoring a systematic extension of the described edge-device concept using frequency analysis before transmitting the data is necessary.

As next steps process-models based on the data will be implemented and tested to overcome the problem of missing physical models for thin foil.

With the availability of detailed production data from the machines valuable information is disclosed to the producer of the machinery and material that can be used in a variety of different scenarios.

References

- [1] N. A. Fleck, K. L. Johnson, Cold rolling of foil, *Journal of Engineering Manufacture*, Vol. 206, 1992
- [2] R. Guo, Recent Advances on Rolling Force Model – Solutions of von Karmann Rolling Equations, Conference Paper – METEC 2015 Düsseldorf, Stahleisen, Düsseldorf, 2015
- [3] Stuart Russell, Peter Norvig, *Artificial Intelligence: A Modern Approach* (Third edition), Pearson, 2009
- [4] Sergey Melnik et al., Dremel: Interactive Analysis of Web-Scale Datasets Proceedings of the VLDB Endowment, Vol. 3, No.1
- [5] Paul McDonald: *Introducing Google App Engine + our new blog*
- [6] Amies, Alex; Sluiman, Harm; Tong, Qiang Guo, Liu, Guo Ning (July 2012), *Infrastructure as a Service Cloud Concepts, Developing and Hosting Applications on the Cloud*, IBM Press, 2012
- [7] Nick Ragouzis et. al., *Security Assertion Markup Language (SAML) V2.0 Technical Overview*, OASIS, 2008
- [8] Wolfgang Mahnke et al., *OPC Unified Architecture*, Springer Verlag, 2009
- [9] Edwin Schicker, *Datenbanken und SQL: Eine praxisorientierte Einführung*, Teubner Verlag, 1996
- [10] unknown, [MS-SMB2]: Server Message Block (SMB) Protocol Versions 2 and 3, MSDN Microsoft, <https://msdn.microsoft.com/en-us/library/cc246482.aspx>
- [11] Lukas Ruebelke, *AngularJS in Action* (1st ed.), Manning Publications, 2015
- [12] Pautasso, Cesare, Wilde, Erik, Alarcon, Rosa (Eds.), *REST: Advanced Research Topics and Practical Applications*, Springer Verlag, 2014
- [13] Markus Hoffman, Ralf Klinkenberg, *RapidMiner: Data Mining Use Cases and Business Analytics Applications*, CRC Press, 2013